Cohort A: https://codeboard.io/projects/385315
Cohort B: https://codeboard.io/projects/385317

All Cohorts:

Task 1 (Guided):

In math, the transpose of a matrix is found by interchanging its rows into columns or columns into rows. For instance given a matrix, **M**:

1	2	3
4	5	6

Its transpose matrix, $\mathbf{M}^T$:

1	4
2	5
3	6

Given an input file, i.e. **matrix.txt**, as shown by figure below:

```
2
4 5
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
2 3
1 2 3
4 5 6

```

The first line, i.e. **2** is the number of matrices that will be transposed, the first line of each matrix contains two space separated integers R and C, the number of rows and the number of columns of the matrix M respectively. Thereafter R lines follow each containing C numbers. Create a java method to transpose the matrix. Please make your code as flexible as possible with different input.

The sample **expected output**:

```
M:
1       2       3       4       5
6       7       8       9       10
11      12      13      14      15
16      17      18      19      20
M':
1       6       11      16
2       7       12      17
3       8       13      18
4       9       14      19
5       10      15      20

M:
1       2       3
4       5       6
M':
1       4
2       5
3       6
```

Task 2 (semi-guided):

Given a **8x8** chess table, with 1 queen and 1 pawn located in random positions. Create a method to check whether the pawn is threatened by the queen or not. Please note that the pawn is threatened by the queen if they are in the same row or column or diagonal.

The input file is: **board.txt**, as shown by the following figure:

```
4
0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 2
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 2 0 0 0 0
1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 2 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
1 0 0 0 0 0 2 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

The first line, i.e. **4**, represents the number of test cases. The following lines are **8x8** chess boards for each test case respectively. The position of the queen is represented by number **1** whereas the pawn is represented by number **2**.

When the pawn is threatened by the queen, you should print **'YES'** and print the chess board, showing the position of Queen and Pawan, and how the pawn is threatened. Otherwise, just simply output **'NO'**.

The sample output is shown below:

```
NO
YES
O  O  O  O  O  O  O  O
O  O  O  Q  O  O  O  O
O  O  O  X  O  O  O  O
O  O  O  X  O  O  O  O
O  O  O  X  O  O  O  O
O  O  O  X  O  O  O  O
O  O  O  X  O  O  O  O
O  O  O  P  O  O  O  O
YES
Q  O  O  O  O  O  O  O
O  X  O  O  O  O  O  O
O  O  X  O  O  O  O  O
O  O  O  X  O  O  O  O
O  O  O  O  X  O  O  O
O  O  O  O  O  X  O  O
O  O  O  O  O  O  P  O
O  O  O  O  O  O  O  O
YES
O  O  O  O  O  O  O  O
O  O  O  O  O  O  O  O
O  O  O  O  O  O  O  O
O  O  O  O  O  O  O  O
Q  X  X  X  X  X  P  O
O  O  O  O  O  O  O  O
O  O  O  O  O  O  O  O
O  O  O  O  O  O  O  O
```

Please make your code as flexible as possible with different input.

Task 3 (unguided):

Cohort A and B:

Ratu Rata Tuli, a stunningly beautiful princess in **Xabrakadabra** kingdom, will travel to **4** of her fellow kingdoms during this Eid Mubarak Eve. Let's say **A, B, C, and D;** She will travel by **Bouraq Airlines**. The cost of the ticket (in million rupiah) from each kingdom to each other, for instance, is shown by the following cost matrix:

	X	**A**	**B**	**C**	**D**
X	0	2	3	**1**	2
A	2	0	**4**	3	4
B	**1**	5	0	2	4
C	4	6	7	0	**2**
D	3	**4**	8	2	0

She has to travel from her kingdom, **X**, visit the other city exactly once, and return back to **X**. Create a method to help the princess find the best travel route in order to get the cheapest ticket.
For instance, if she chooses the route: **X-C-D-A-B-X** it will cost **1+2+4+4+1=12** million rupiah. But she still wonders whether there is a cheaper route or not.

Given, a sample input file, **costmatrix.txt**, as shown below:

```
3
4
0 2 3 1 2
2 0 4 3 4
1 5 0 2 4
4 6 7 0 2
3 4 8 2 0
4
0 2 3 1 9
2 0 4 3 4
1 1 0 2 4
1 3 1 0 2
3 4 1 2 0
5
0 2 3 1 1 3
2 0 4 3 4 1
1 2 0 2 4 2
1 1 1 0 2 1
3 4 1 2 0 4
1 2 3 4 1 0
```

The first line, i.e. **3** is the number of test cases, the first line of each test case contains an integer **S**, the number of kingdoms that Ratu Rata Tuli will visit. Thereafter **S+1** lines follow representing the cost matrix.

The sample expected output is shown below:

```
Test Case :1
  X A B C D
X 0 2 3 1 2
A 2 0 4 3 4
B 1 5 0 2 4
C 4 6 7 0 2
D 3 4 8 2 0
Best Route:X--C--D--A--B--X (12)
Test Case :2
  X A B C D
X 0 2 3 1 9
A 2 0 4 3 4
B 1 1 0 2 4
C 1 3 1 0 2
D 3 4 1 2 0
Best Route:X--C--D--B--A--X (7)
Test Case :3
  X A B C D E
X 0 2 3 1 1 3
A 2 0 4 3 4 1
B 1 2 0 2 4 2
C 1 1 1 0 2 1
D 3 4 1 2 0 4
E 1 2 3 4 1 0
Best Route:X--C--E--D--B--X (5)
```

Please make your code as flexible as possible with different input.